NASA TECHNICAL TRANSLATION NASA TT F-13,873

USING NATURAL AND ALGORITHMIC REDUNDANCY IN INFORMATION SYSTEMS AND AUTOMATIC CONTROL SYSTEMS

G. A. Kozlik and Yu. G. Savchenko

N72-14218 (NASA-TT-F-13873) USING NATURAL AND ALGORITHMIC REDUNDANCY IN INFORMATION SYSTEMS AND AUTOMATIC CONTROL SYSTEMS G.A. Kozlik, et al (Scientific Translation Service) Aug. 1971 12 p CSCL 09B G3/08

Unclas 12021

Translation of: "Ob ispol'zovanii yestestvennoy signal'noy i algoritimicheskoy izbytochnosti v informatsionnykh i avtomatizirobannykh sistemakh upravleniya", Chetverty Simposium po Probleme Izbytochnosti v Informatsionnykh Sistemakh, (Fourth Symposium on the Problem of Redundancy in Information Systems), Leningrad, USSR Academy of Sciences, 1970, pp 355-364

13873

FACILITY FORM 602

12 (ACCESSION NUMBER) (PAGES) (NASA CR OR TMX OR AD NUMBER)

G3 (THRU) (CODE)

08 (CATEGORY)

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546 AUGUST 1971

USING NATURAL AND ALGORITHMIC REDUNDANCY IN INFORMATION SYSTEMS AND AUTOMATIC CONTROL SYSTEMS

G. A. Kozlik and Yu. G. Savchenko

ABSTRACT. The parameters of a computing system are given as a complex of control algorithms. The necessary control quality can be obtained by the proper organization of the input information, the structure of the algorithmic control system, and the computing process [1].

Here, by algorithmic control system (ACS), we mean the functional, algorithmic, and informational makeup of the automatic control system, which allows the given complex of control algorithms in the computing systems to interact and output the necessary control commands (recommendations). These commands are usually uniquely determined by the organization and the current state of the ACS, and also by the input information and computing system (CS) parameters. /355*

Control quality is specified by parameters, such as reliability, accuracy, and computing time, which are to be held within certain limits. The use of redundancy at various levels is one of the effective methods for improving these performance indices [2, 3].

This article considers questions of using the natural informational and algorithmic redundancy in automatic control systems for commercial use. /356

1. The reliable operation of information systems, which contain electronic digital computers (EDC) and a multi-branched information-collection network,

*Numbers in the margin indicate the pagination in the foreign text.

is determined by the magnitude of the EDC errors and errors in the input apparatus, the noise level in the data-transmission channels, and the frequency of input errors that are made by the operator. These error components differ physically, and each has a different weight in a specific case. A series of well-known methods (information-coding with error correction, apparatus control and EDC structural redundancy, control checks, etc.) can significantly reduce the effects of the first two types of errors. Until recently, relatively little attention was given to operator errors, although operational experiments have shown that they have a significant weight in the overall balance.

Operator errors cannot be corrected by the usual methods of error-correcting coding, since these errors essentially occur in the information symbols, which are input to the coding device and which are thus not covered by the abilities of the code. Attempts to have the operator do the coding, as well as the actual input, do little to improve the situation. The additional calculations, which the operator must perform (for the input) slow the input rate and introduce new errors that result from the increased load on the operator.

On the other hand, the initial information from the collection points in an operational information system arrives at the EDC in an unprocessed form. That is, in most cases the data are not subjected to any preliminary processing. As a rule, this is done because there is no special apparatus for preliminary processing of the information at the collection location. More important, such processing is not economically expedient when there are powerful central computers. Under such conditions, the input information contains significant informational redundancy in most cases. For example, the transmitted data /357 consist of numbers, which correspond to the number of finished parts at some working station. Let this number vary within the limits of 200-300 pieces. Representing these numbers usually requires three decimal or ten binary digits. As a result, only 100 out of 2^7 (128) possible combinations are used. That is, the natural redundancy of the data is ten-fold (the ratio of the number that is used). Such a situation is typical for most information flows in

industrial automatic control systems. We shall see what possibilities arise, when the data generated by the information source contain natural informational redundancy.

Let there be some source of binary information, such as an input device, an output-transmission channel, etc. We denote the bits by Y_1, Y_2, ..., Y_K, where K is the number of bits in the binary word. Let the number of combinations used for the data source by $N < 2^K$. This means that only a part of the number of all possible combinations is being used. We form two subsets out of all possible words:

$$U_p = \{y_1, y_2, \ldots, y_\alpha, \ldots, y_N\}$$

for solution words and

$$U_3 = \{y'_1, y'_2, \ldots, y'_\beta, \ldots, y'_{2^K-N}\}$$

for unused or forbidden words.

We now turn to actual errors. Obviously, any error which arises either from any malfunction of the input device or from channel noise (or inattention from the operator who performs the input operations) transforms some correct word Y_α into a distorted word — for example, Y_γ. Such a transformation corresponds to a line on a diagram, whose nodes are the words Y_α and Y_γ. The diagram completely determines the error, which has the character of an error vector $E = Y_\alpha \oplus Y_\gamma$.

If the word Y_γ belongs to the subset U_z, then the error can easily be observed and corrected. Actually, if it is known beforehand that the appearance of the word Y_γ can only result from an error in the word Y_α, then it is sufficient to identify any appearance of the word Y_γ with Y_α. In this case, the very fact that the word appears means that there is an error (this is

/358

equivalent to observing the error), and the identification procedure results in a correct word (correction).

However, it certainly is not always so easy to correct errors. In the first place, not one, but a combination of errors, affects the information source at the outputs. Secondly, the errors do not necessarily transform a solution word into a forbidden word. Cases are possible where an error in one solution word results in a word which is also a solution word.

The following can be shown for an arbitrary combination of error vectors:

1. If no line on the graph leads to a node belonging to U_p, then any error given by the graph can be observed.

2. If no more than one line leads to every node corresponding to words from the subset U_z, and no line leads to a node corresponding to words from the subset U_p, then any error on the diagram can be corrected.

Actually, if the error graph (or diagram) has these properties, then any error can be corrected by identifying the nodes which are connected by each line, if the error is caused by a transition to a node belonging to U_z. This proves that the formulated conditions are sufficient.

We now consider these properties in the terminology of coding theory. The essence of error-correcting coding consists of adding an alphabet of additional conditions to the set of words. Decoding (correcting) errors is accomplished by verifying that these conditions are fulfilled. If there are no errors, then all conditions are fulfilled, and the obtained information is considered to be correct.

If some errors arose (from the number of words that can be corrected with the conditions that are introduced), then some of the conditions will not be fulfilled. The composition of the fulfilled conditions determines the presence /359

and location of the errors in a given word.

In its most general form, the operation of checking that the conditions are fulfilled may be represented by the control system

$$r_i = f_i(y_1, y_2, \ldots, y_K), \quad i = 1, 2, \ldots, m, \qquad (1)$$

where r_i is the result of the i^{th} check; $y_1, y_2, \ldots, y_K$ are the symbols in the word; and m is the number of conditions.

For binary information, it is natural to take the $y_1, y_2, \ldots, y_K$ and r_i as binary variables and the $f_i(y_1, y_2, \ldots, y_K)$ as Boolean functions.

For exactness, we will consider that $r_i = 0$ if the i^{th} check is fulfilled, and that $r_i = 1$ if it is not.

Observing errors. First of all, we ascertain what information should be carried in the result, or combination of results, of the checking operation. Obviously, if we are only observing errors, then it is sufficient to know whether the word contains an error or not. This information can be obtained from one binary digit. In other words, any combination of errors can be observed with one check, that is m = 1.

Obviously, observing an error requires that the single checking function $r = f(y_1, y_2, \ldots, y_m)$ equal 1 for sets corresponding to the subset U_z, and that this subset is the only necessary condition for observing errors of an arbitrary type. Moreover, the formulation of this condition is constructed such that in essence it gives all the constants of the Boolean function $f(y_1, y_2, \ldots, y_m)$ and also completely defines it.

Correcting errors. The number of necessary checks for correcting errors depends on the decoding procedure that is used. We will consider a more convenient procedure, which is used for decoding group codes. The word /360

obtained from the output of the information source is subjected to m checks. The combination of checking results (syndrome) shows the location of the error, that is, which bits of the word have been changed. The actual correction consists of inverting these symbols. Obviously, correcting any one of the errors requires the ability to make a checking sequence $R = r_1, r_2, \ldots, r_m$ correspond with each error vector E. If the number of correcting vectors is M, then the result is

$$2^m \geq M+1, \quad \text{or} \quad m \geq \log_2(M+1), \tag{2}$$

where a 1 is written as a result of a null check in the sequence to indicate an absence of errors.

Generally speaking, the value of m from Equation 2 can be reduced significantly for other decoding procedures. For example, if an error vector E_i "affects" only one word Y_α, and the resulting distorted word $Y_\gamma = Y_\alpha + E_i$ cannot arise as the result of any other error, then, as we have already remarked, it is essentially sufficient in this case only to observe the error. If the whole set of correctable errors has the same property, then obviously m = 1.

However, we turn to a conventional decoding procedure. In this case, we are given a set of incorrectable errors, as well as a set of words which are used by the information source. Then it is convenient to utilize the properties of nonlinear codes to determine the checking conditions. Nonlinear codes can correct a list of errors [4]. To do this we write out the selected checking sequence, along with each error vector. If some of the digits in the checking sequence are equal to 1, then we will say that the check r_j checks the error vector E_i. If $r_j = 0$, then the check does not check the vector E_i. In this terminology it is possible to formulate the following obvious rules for selecting the checking functions: each of the checking functions r_j should equal 1 for sets which correspond to $y_\eta + E_j$, where E_j is the error vector checked by r_j and y_η is the word affected by E_j; r_j should equal 0 for sets

that correspond to words $Y_\rho + E_n$, where E_n is an error vector that is not checked by r_j. /361

Having written all the words, which are obtained by superimposing error vectors which are checked by a given r_j, we obtain all sets for which the checking function equals 1, that is, we obtain all unit constants of the desired function.

In some specific cases it is expedient to obtain the checking conditions from the natural limits, which are imposed on the composition of the data at the collection point. For example, the input data on production information is often input into the information system as an increasing sum, that is, there is a condition of the form

$$Y_{i+1} > Y_i$$

where the index i determines the sequence of the words in time. If the number of this type of condition is sufficiently large, then a corresponding correction procedure can be formulated. In many cases such a procedure can be more useful than computing the values of Boolean functions, from the point of view of machine utilization.

2. The objectives of automation are becoming increasingly complex, and require increasingly powerful and complex information systems and automatic control systems. Thus, the role of investigations into methods that efficiently utilize computing systems is also increasing. The development of design and construction methods for ACS's makes it possible to reach this goal to a significant degree.

So-called "utility" algorithms and programs for solving planning and economic problems, production-distribution problems, and information problems have been introduced into the mathematics of commercial automatic control systems (ACS). These algorithms include algorithms for organizing masses

of information (sorting, forming documents), optimal planning (methods for mathematical program-planning and combinatorial analysis), information and reference service to subscribers, etc. /362

However, the efficient use of the computing system as a general-purpose information-processing system depends significantly on the presence of so-called "general mathematical operations" in the composition of the ACS. These operations consist of a set of special groups of redundant algorithms and programs (with respect to functional-utility algorithms and programs).

First, there is a group of test programs, which are designed for complex checking of the state of various units and devices in the CS and for holding the operating system within normal conditions. This group also contains programs for continuously checking the computation process, for analyzing the problem solutions for signs of errors, and for automatically repeating the solutions, if this is necessary. This group should include all algorithms for checking input information and for providing a reliable tie to the "human" system.

Furthermore, there is a group of operational programs, which perform control functions for timing system operations, organizing information interchange, establishing the order of data (problem) acceptance, and running these problems in the ACS according to a priority system. These algorithms also increase equipment efficiency by simultaneously operating several devices, etc.

For actual systems, this group of redundant algorithms can include program-translation algorithms with various input languages and interpretive systems with standard program libraries.

Finally, the normal operation of the ACS is improved by a group of auxiliary programs in the system — for example, service programs for communication between the operator and the CS.

We illustrate these algorithms with an example. /363

Suppose that we want to find the values of m parameters $x_1, x_2, \ldots, x_m$ which maximize (or minimize) some industrial quality control function:

$$K = K(X_0, X_1, U, \Theta)$$

under conditions that satisfy equations and inequalities which express the production relationships and limitations:

$$F_i[R, X, U, \Theta] = 0; \quad i = 1, 2, \ldots, i_0;$$
$$\Psi_\lambda[R, X, U, \Theta] \leqslant = \geqslant 0; \quad \lambda = 1, 2, \ldots, \lambda_0,$$

Reproduced from best available copy.

Here R is the set of production resources (material, labor, energy, etc); X, X_0, X_1 are the set of system control parameters at an arbitrary time, the initial time, and final time, respectively; U is the set of control algorithms; Q is the combination of parameters that characterize outside influences.

Obviously, it is currently impossible to solve this problem for complex control systems at the level of industrial interprises. The large dimensions of the model and the complex nonlinear relationships between various parameters make the computing problems insurmountable. It is possible to escape this situation by introducing an additional group of algorithms. These algorithms decompose the algorithmic model into a hierarchy of subsystems, according to a criterion that minimizes the "optimization loss" which results from separating the relationships of the various parts, from changing the operating rules, and from introducing algorithmic and informational redundancy by expediently doubling the separate algorithms (or blocks of algorithms) and information masses in various subsystems, etc.

The extent to which algorithmic and informational redundancy can be introduced and the ability to use this redundancy in automatic control systems are determined by the requirement that the problem be formulated correctly /364